Learning Spatial Relations with a Standard Convolutional Neural Network

Kevin Swingler[a] and Mandy Bath
Computing Science and Mathematics, University of Stirling, Stirling, FK9 4LA, Scotland

Keywords: Convolutional Neural Networks, Spatial Reasoning, Computer Vision.

Abstract: This paper shows how a standard convolutional neural network (CNN) without recurrent connections is able to learn general spatial relationships between different objects in an image. A dataset was constructed by placing objects from the Fashion-MNIST dataset onto a larger canvas in various relational locations (for example, trousers left of a shirt, both above a bag). CNNs were trained to name the objects and their spatial relationship. Models were trained to perform two different types of task. The first was to name the objects and their relationships and the second was to answer relational questions such as "Where is the shoe in relation to the bag?". The models performed at above 80% accuracy on test data. The models were also capable of generalising to spatial combinations that had been intentionally excluded from the training data.

1 INTRODUCTION

The ability to understand the relationships between objects in an image is an important step towards a complete solution to computer vision. Objects can have many types of relationship, such as subject-object or relative location. This paper describes work aimed at allowing neural networks to learn about the spatial relationships between pairs of objects in an image and report both image labels and relationship labels, for example "The bag is above the coat". This work is part of a project that aims to use computer vision to build assistive technology for the blind. The project is called the Artificial Intelligence Sight Loss Assistant (AISLA)[1] and this work contributes to a module designed to process simple questions about the locations of objects in a room.

Convolutional Neural Networks (CNNs) have proven to be very successful in recent years at a number of signal processing tasks including computer vision. Early CNNs were designed to classify a whole image and so were only able to process a single object at a time. Work on improving these algorithms continues. Examples include LeNet (LeCun et al., 1998), AlexNet (Krizhevsky et al., 2012), VGG (Simonyan and Zisserman, 2014), the Inception models (Szegedy et al., 2015), (Szegedy et al., 2016), ResNet (He et al., 2016), and ResNeXt (Xie et al., 2017). At the same time, a number of large image collections have been published with data identifying the objects in the images and their locations. Examples include the COCO dataset (Lin et al., 2014), the Fashion-MNIST dataset (Xiao et al., 2017) and the ImageNet database (Deng et al., 2009).

The same convolutional idea is also applied to the dual task of locating and labelling more than one object in an image. This is collectively known as object detection and well known object detection models include R-CNN (Girshick et al., 2014), Faster R-CNN (Ren et al., 2015), and the series of YOLO models (Redmon and Farhadi, 2017), (Redmon and Farhadi, 2018). Object detectors generate a list of object labels and associated bounding boxes, locating them in the image. Given the bounding boxes in an image, some simple geometric calculations can be used to test the relative positions of objects to each other in the flat plane of the image. However, it is an interesting question as to whether a CNN, with its focus on local features, is capable of learning larger scale spatial relationships among the objects in an image.

Some work has been carried out that attempts to describe spatial relationships in images. Automated image captioning uses a mixture of natural language processing and computer vision to associate an image with a descriptive sentence such as "A man walking on a beach with a dog". Some of the descriptions contain spatial relationships - "A vase on a table" for example, but as the words are generated statistically,

[a] https://orcid.org/0000-0002-4517-9433
[1] https://www.aisla.org.uk/

Swingler, K. and Bath, M.
Learning Spatial Relations with a Standard Convolutional Neural Network.
DOI: 10.5220/0010170204640470
In *Proceedings of the 12th International Joint Conference on Computational Intelligence (IJCCI 2020)*, pages 464-470
ISBN: 978-989-758-475-6

the spatial meaning is lost. Many of these models mix CNNs with LSTMs to process the series of words that make up a caption, see (Wang et al., 2016) for example. Hossain et al. (Hossain et al., 2019) provide a useful review of image captioning techniques.

In 2014, in his talk on what is wrong with convolutional neural nets, Geoff Hinton talked about the limitations of max pooling and how CNNs can recognise the right elements, but in the wrong order. For example, it might detect two eyes, a nose and a mouth and classify a face even if those elements are not arranged as a face. We were interested in whether or not a CNN could be made to learn such spatial relationships if the target outputs made them explicit. This is slightly different from the point that Hinton was making, but sparked the question all the same. Can a CNN learn image wide spatial relationships by simply encoding the name of such relationships at the output layer?

There have been several attempts at explicitly addressing the challenge of learning spatial relationships among objects in an image using an architecture that adds spatial specific elements to the standard CNN. Mao et al. (Mao et al., 2014) use a mixture of recurrent network layers and convolutional layers in a multimodal approach they call an m-CNN. The model uses a statistical approach to produce words that form sentences that describe images. Words are selected from a probability model based on the image and the previous words in the sentence.

More recently, Raposo et al. (Raposo et al., 2017) proposed relation networks (RN) as a way of allowing networks to learn about the relationships between objects in a scene. The RN models the relationships between pairs of objects and is used in conjunction with CNNs and LSTMs for image and language processing.

In this paper, we address the question of whether a simple CNN architecture without recurrent or LSTM components is able to learn relative spatial relationships in images. The important question is whether or not a CNN can learn to generalise concepts such as *above* or *below* from example images without an architecture that is specifically designed to capture those relationships. This was done by generating images with a small number of object classes arranged in a variety of spatial configurations while ensuring that some combinations did not appear in the training data. When tested, the model was able to correctly report the relative locations of object combinations that were absent from the training data.

The motivation for the work is to add specific output nodes to a CNN, which refer to a defined concept. In this case, the concepts describe relative locations, but in future work they might describe an action or even an intention. Rather than generating a sentence (such as the girl is drinking the milk) that requires further post-processing to extract meaning, we aim to generate meaningful outputs directly from the image (object=girl, subject=milk, verb=drink, for example). We want to be able to use a single architecture (a standard CNN) and change only the output targets to be able to train on different meaningful relationships among objects in an image.

The remainder of this paper is organised as follows. Section 2 describes the preparation of the training, validation and test data. Sections 3 and 4 describe two experiments with CNNs for spatial relation recognition. Finally, section 5 provides an analysis of the results and some ideas for further work.

2 DATA PREPARATION

The datasets for the experiments were constructed using the Fashion-MNIST dataset (Xiao et al., 2017). Images in this collection are 28 by 28 pixels in size and they were used to generate larger images of 56 by 56 pixels by pasting two or three of the original images onto a larger canvas. The original images were selected at random without replacement and placed on the larger canvas in randomly chosen non-overlapping locations. In this way, any two objects in an image could have a clearly defined spatial relationship from the set {above, below, left, right, above left, below left, above right, below right}. The data was automatically labelled using the algorithm that generated it, making the process of producing large quantities of data very efficient. Figure 1 shows two example images. Note that the Fashion-MNIST images are 28 by 28 pixels, so the figures in this paper represent the quality of those images accurately.

The Fashion-MNIST object classes are: T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot. There are 60,000 training images and 10,000 test images in the dataset. All of the images are provided in grey scale so they do not have a colour dimension.

The training/validation/test protocol was as follows. The original MNIST training data were split into 20% test data and 80% training data before the composite images were generated. The 80% used for training were further split into 80% train and 20% validation sets, all before the composite images were created. Consequently, the train, validation and test data share no original images in common. What is more, on different training runs, certain combinations of object class and spatial relationship (shirt above bag,

Figure 1: Two example training images. The first showing a sneaker below left of a pullover and the other showing a t-shirt below left of a dress. The categories were defined so that objects were either perfectly aligned, meaning that one is to the left of the other, or offset vertically meaning one is below left or below right the other.

for example) were actively excluded to ensure that the validation and test data contained novel combinations. In total there were 48,000 training examples, 12,000 validation examples and 10,000 test examples created. The validation data were used to tune the hyperparameters of the CNN and the test data were used to report the final accuracy values.

Two different experiments were performed, each on a different variant of the spatial relationship labelling task. The first trains a model to produce the class names of two objects in an image along with the name of the spatial relationship between them. The second takes an image with three objects in it along with the names of two of them and produces the name of the relationship between the two named objects as output. The experiments are described in the next two sections.

3 EXPERIMENT ONE - TWO OBJECTS

This section describes the first experiment, in which a CNN is trained to describe an image containing two objects. The images in the training data each contain two objects from the fashion MNIST data set and the target output for each consists of the two labels plus the name of the spatial relationship between them.

Each image forms the input part of a training point and the output is a vector of 24 binary values. The first 10 form a one-hot encoding of the name of the first object in the image. The second 10 use the same representation for the second object in the image and the final four represent the four possible spatial relationships of the first object to the second. They are in the set {left, above, below left, and below right}. There is no need to encode the relationships to the right of or below as they can be represented by switching the order of the object labels. For example, a coat to the left of a bag would encode coat, bag, left and a coat to the right of a bag would encode bag, coat, left. Consequently, each output vector contains exactly three values set to one and the rest at zero.

3.1 Network Architecture

A convolutional neural network architecture was used as follows. The input layer takes grey level images of 56 by 56 pixels, so has a volume of $56 \times 56 \times 1$. There are then three convolutional layers, the size of which is explored as one of the architectural hyperparameters. Each one of the three layers uses batch normalisation, max pooling and dropout, all optimised during the training and validation process. All layers except the output layer use ReLU activation functions as this is well established in the literature as a good choice for speed and stability.

There is then a fully connected layer that is itself fully connected to the final, output layer, which uses logistic activation functions paired with a cross entropy loss function. Softmax activation here would not be suitable as there are always three nodes with a target value of one, and softmax forces all the outputs to sum to one.

During training, the hyperparameters were tuned using a mixture of manual and hyperband search (Li et al., 2017). The hyperparameters, their possible settings and search method are shown in Table 1. The manual search involved the researcher making a judgement about how to set the hyperparameter based on previously built models from the same set of experiments. It is fair to say that the model was not fully optimised, but the purpose of the work is to test whether or not a CNN can learn spatial relationships rather than squeezing the last percentage of accuracy from the data.

Table 1: The hyperparameter search space for the convolutional neural network. In the Search column, H means hyperband and M means manual.

Hyperparameter	Options	Search
Number of filters	32, 64, 128	H
Kernel size	3×3, 5×5, 11×11	M
Optimiser	Adam, SGD	H
Learning rate	0.01, 0.001, 0.0001	M
LR Decay	Yes, No	M
Batch size	1, 32, 64, full	M
Dropout	0.2, 0.3, 0.4, 0.5	M

During training, early stopping was used if the validation accuracy stopped falling. The first two convolutional layers used small kernels, as is common in the literature, but for the third layer we experimented with larger kernels to allow the network to capture

broader spatial relationships. Peng et al. (Peng et al., 2017) describe the advantages of larger kernels when performing object localisation. Learning rates started at the values stated in Table 1, but learning rate decay was also found to be advantageous.

3.2 Hyperparameter Search Results

The results of the hyperparameter search were a model with the architecture shown in Figure 2. The model was trained for 162 epochs before early stopping. The optimal hyperparameters were found to be a learning rate of 0.001 with learning rate decay on each epoch. The optimal batch size was found to be 32 and the dropout rate was 0.3. ReLU activation functions were used throughout except at the output layer, where logistic functions were used. Adam was found to be more effective than stochastic gradient descent (SGD) as the optimiser.

It was found that using smaller kernels at the final convolutional layer adversely affected the accuracy of the model, suggesting that the larger kernel is needed to capture positional relationships. This is not a surprising finding as these relationships naturally occur over large distances. Being at later layers, the larger kernels are able to operate on smaller input volumes than the full input image while still spanning the full scope of the original image. This is because the architecture of the network shrinks the input size at each layer with max pooling so a single cell in layer three covers the information in 4 pixels of the original image.

3.3 Results

After 162 training epochs, early stopping caused the process to terminate. The training accuracy was 0.93 and the test accuracy was 0.83. The specific combination of the objects bag and trousers were excluded from the training data but included in the test data. The test data images containing a bag and a pair of trousers were then used to test the model's ability to generalise to combinations of object and location that were not in the training data. The model was able to correctly give the object labels and relative locations for these images. An example image is shown in Figure 3.

4 EXPERIMENT TWO - THREE OBJECTS

In the first experiment described above, the task for the CNN was to name the classes for two objects in

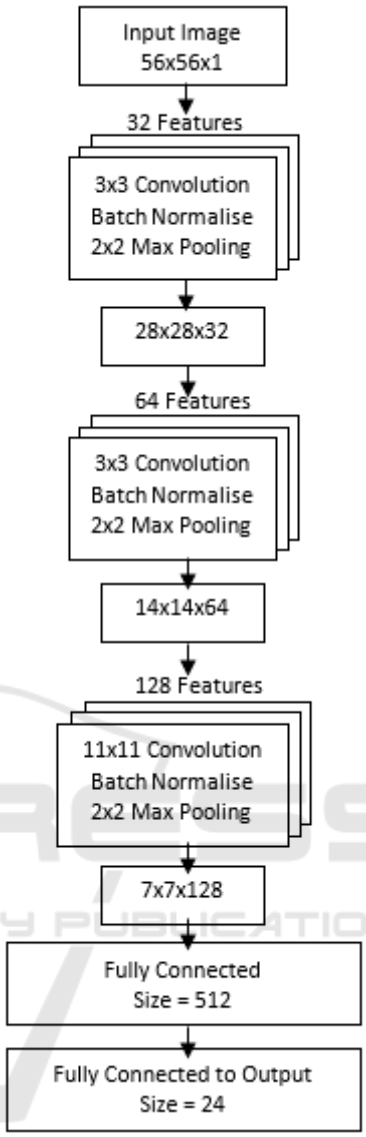

Figure 2: The architecture of the relationship detector CNN.

Figure 3: An example output from the model showing the correct response to an image containing a pair of trousers to the left of a bag. This was a combination of objects that was explicitly excluded from the training data.

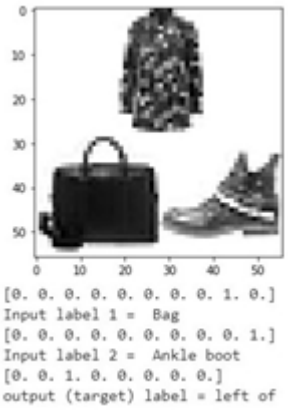

Figure 4: An example input and output encoding. In this example, the relationship "bag left of ankle boot" is represented.

an image and label the spatial relationship between them. In the second experiment, the task is to take an image containing three objects, along with the labels for two of them as an input and generate the nature of the relationship between them as the output. This is equivalent to answering a question like "Where is the coat in relation to the bag?". It presupposes that you know the identity of two of the objects in the image. However, the presence of a third object in the image means that this is not simply a question of identifying where the objects are. The network must identify which are the two objects of interest and then generate the relationship between them. Figure 4 shows an example input image.

The data representation for this task consists of the image as a 56×56 array of grey level pixel values plus two vectors of ten, each with a one-hot encoding of the object class to include in the question. We do not attempt to process whole sentences, such as "What is to the left of the bag?", but encode the question explicitly as two words and an image. The output represents the 8 possible relationships of the second object to the first from the set {above, below, left, right, below left, below right, above left, above right}. As with the inputs, we are not trying to generate full sentences, just the answer to the encoded question. Of course, it is trivial, given the two input object labels and the name of their relationship to generate a meaningful sentence from the output.

Training images were generated by placing three random images from the Fashion-MNIST training data in non-overlapping locations on a 56×56 pixel blank canvas. Two of the objects were selected at random and their spatial relationship calculated and encoded as the target output. Note that each training image represents only one of the possible relationships it contains. In figure, 4, for example, the coat object is not represented in the output.

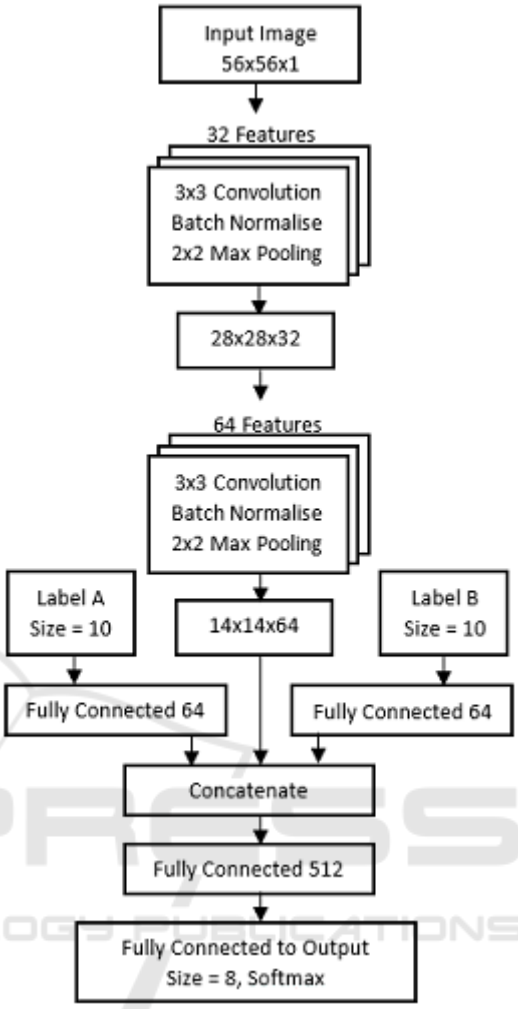

Figure 5: The architecture of the three object CNN.

The architecture for the model has three channels that merge in the first fully connected layer. One uses the standard CNN components of convolutions, max pooling, ReLU, and batch normalisation. The other two channels simply feed the one-hot encoded representations of the question words into the first fully connected layer. This layer then merges the three channels into a final, fully connected layer to a one-hot encoded softmax layer that represents the target relationship label at the output. The architecture is shown in Figure 5.

4.1 Results

After a hyperparameter search following the protocol described in section 3.1, a network with the architecture shown in Figure 5 was trained until early stopping at 100 epochs. The learning rate was initially

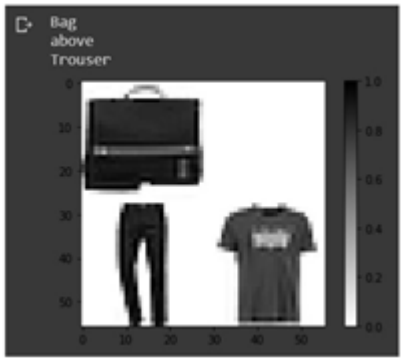

Figure 6: The model correctly labelling a bag above the trousers. The training data contained no examples of this combination.

0.001 and rate decay was used to decrease it further after each epoch. Convolutional layers had ReLU activation functions and the output was softmax. The dropout rate was 0.3 and Adam was used for the optimiser. Categorical cross entropy was used for the cost function.

The final model achieved an accuracy of 94% on the training data and 86% on the unseen test data. As before, images containing a bag and a pair of trousers were missing from the training data, but were explicitly included for testing. Figure 6 shows an example of the model correctly generalising to generate an answer to the question "Where is the bag in relation to the trousers?".

5 CONCLUSION

This short piece of work shows that a standard convolutional neural network is capable of learning image-wide concepts about the relative locations of objects in an image. The architecture of the CNN uses a combination of max pooling layers that shrink the input dimensions at each step and a large kernel at the final layer. This allows broad scale relationships across the original image to be captured in smaller feature maps. We speculate that the large kernel at the final convolutional layer of the network is responsible for learning the relative locations. There is evidence for this in the fact that reducing the size of the final kernels reduces the ability of the network to correctly label relative locations of objects in an image.

The second of the two experiments shows that a very simple form of visual question and answering can be implemented with a standard CNN by encoding the question at the inputs and the correct answer at the outputs. The input encodes both the image and the question, which specifies which elements of the image should be used to generate the answer. So the question "Here is a picture that contains a bag, a coat and a shoe, but where is the shoe in relation to the bag?" can be answered with a one-of-eight encoding of relative locations.

There is plenty of scope to extend this work. Larger images with more objects could be introduced and the images could be placed on a noisy background rather than a plain white one. Real images could also be used. For example, the well known COCO dataset (Lin et al., 2014) has object labels and bounding boxes in its annotations. The relative locations of objects could be roughly inferred from the bounding box coordinates. Other relationships types could also be introduced, for example "on" as in "the lamp is on the desk" or "in front" or "behind".

The primary application that motivates this work is the development of a personal assistant technology for people with sight loss. Fixed cameras in a home could be used to answer location based questions such as "where did I leave my radio?" or to warn a user about a potential obstacle or danger. Other applications that require a specific representation of relative locations include self-driving cars and warehouse picking robots.

The difference between this approach, where specific relationships are one-hot encoded as specific nodes, and the sentence generating approaches described in section 1 is that the relationship is made explicit in the representation. With the three labels: "Coat", "Left", "Bag", one can easily generate a caption: "The coat is to the left of the bag" but one can also answer other questions such as "What is to the left of the bag?" or "Where is the coat?" without the need to use natural language understanding to decode an automatically generated caption. The *facts* of the image are made explicit.

REFERENCES

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hossain, M. Z., Sohel, F., Shiratuddin, M. F., and Laga, H. (2019). A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)*, 51(6):1–36.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Mao, J., Xu, W., Yang, Y., Wang, J., and Yuille, A. L. (2014). Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*.

Peng, C., Zhang, X., Yu, G., Luo, G., and Sun, J. (2017). Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361.

Raposo, D., Santoro, A., Barrett, D., Pascanu, R., Lillicrap, T., and Battaglia, P. (2017). Discovering objects and their relations from entangled scene representations. *arXiv preprint arXiv:1702.05068*.

Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Wang, C., Yang, H., Bartz, C., and Meinel, C. (2016). Image captioning with deep bidirectional LSTMs. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 988–997.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.